Granul. Comput. (2017) 2:357–386
DOI 10.1007/s41066-017-0049-2

ORIGINAL PAPER

Semi-random partitioning of data into training and test sets in granular computing context

Han Liu[1] · **Mihaela Cocea**[2]

Received: 26 May 2017 / Accepted: 26 July 2017 / Published online: 9 August 2017

Abstract Due to the vast and rapid increase in the size of data, machine learning has become an increasingly more popular approach for the purpose of knowledge discovery and predictive modelling. For both of the above purposes, it is essential to have a data set partitioned into a training set and a test set. In particular, the training set is used towards learning a model and the test set is then used towards evaluating the performance of the model learned from the training set. The split of the data into the two sets, however, and the influence on model performance, has only been investigated with respect to the optimal proportion for the two sets, with no attention paid to the characteristics of the data within the training and test sets. Thus, the current practice is to randomly split the data into approximately 70% for training and 30% for testing. In this paper, we show that this way of partitioning the data leads to two major issues: (a) class imbalance and (b) sample representativeness issues. Class imbalance is known to affect the performance of many classifiers by introducing a bias towards the majority class; the representativeness of the training set affects a model's performance through the lack of opportunity for the algorithm to learn, by not presenting it with relevant examples—similar to testing a student on material that was not taught. To solve the above two issues, we propose a semi-random data partitioning framework, in the setting of granular computing. While we discuss how the framework can address both issues, in this paper, we focus on avoiding class imbalance when partitioning the data, through the proposed approach. The results show that avoiding class imbalance results in better model performance.

Keywords Granular computing · Machine learning · Data partition · Multi-granularity learning · Class imbalance · Sample representativeness

1 Introduction

Machine learning is a branch of artificial intelligence, which is increasingly used in the big data era for the purpose of knowledge discovery and predictive modelling. The former purpose generally means that a model is learned from data and some previously unknown patterns can be extracted from the model (Liu et al. 2016). The latter purpose means that a model is learned from data and the model is then used to predict on any new data instances. For both knowledge discovery and predictive modelling, it is essential to partition a data set into a training set and a test set (Liu et al. 2016). In particular, for the purpose of knowledge discovery, the training set is used for a machine learning algorithm to discover any new patterns, and the test set is then used to validate the degree to which the patterns truly exist and are trustable. In contrast, for the purpose of predictive modelling, the training set is used for a machine learning algorithm to build a model, and the test set is then used to evaluate against the predictive accuracy of the model.

✉ Han Liu
LiuH48@cardiff.ac.uk

Mihaela Cocea
mihaela.cocea@port.ac.uk

[1] School of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Cardiff CF24 3AA, UK

[2] School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace, Portsmouth PO1 3HE, UK

In the context of partitioning a data set into a training set and a test set, it has been critical to decide effectively on which part of the data set is selected as the training set, and which part is selected for the test set (Liu et al. 2017). In the traditional machine learning, it is a normal practice that researchers and practitioners choose to do the data partitioning in a fully random way. This way of partitioning, however, leads to two major issues: (a) class imbalance and (b) sample representativeness issues.

The first issue of class imbalance (Longadge et al. 2013; Ali et al. 2015) is known to affect many classifiers' performance (Sotiropoulos and Tsihrintzis 2017). Randomly partitioning the data, however, can lead to class imbalance in the training and the test set, even when there is no imbalance in the overall data set. For example, let us consider a 2-class (e.g., positive class and negative class) data set with a balanced distribution of instances across classes, i.e., 50% of the instances belong to the positive class and 50% of the instances belong to the negative class. When the data set is partitioned by selecting training/test instances randomly, it is likely that the class balance of the data set will be broken, which would lead, for example, to more than 50% of the training instances belonging to the positive class and more than 50% of the test instances belonging to the negative class, i.e., the training set has more positive instances than negative ones, while the test set has the opposite situation.

The second issue is about sample representativeness and the fact that the random partitioning may lead to high dissimilarity between training and test instances. In the context of student learning, the training instances are like the revision questions and the test instances are like the exam questions. To test effectively the performance of student learning, the revision questions should be representative with respect to the learning content covered in the exam questions. The random partitioning of data, however, can result in the case that the training instances are dissimilar to the test instances, which corresponds to the situation that students are tested on what they have not yet learned. Such a situation not only leads to a poor performance, but also to a poor judgment of the learner capability. Thus, in the context of predictive modelling, some algorithms may be judged as not being suitable for a particular problem due to a poor performance, when in reality the poor results are not due to the algorithm, but to the representativeness of the training sample.

To address the two issues mentioned above, we propose, in this paper, a semi-random data partitioning framework in the setting of granular computing, towards effective selection of training and test instances. In particular, we focus on dealing with the class imbalance issue and provide a brief proposal towards dealing effectively with the sample representativeness issue.

The rest of this paper is organized as follows: Sect. 2 provides theoretical preliminaries on data partitioning and granular computing concepts. In Sect. 3, we present a multi-granularity data partitioning framework for controlling effectively the partitioning of data into a training set and a test set, towards overcoming the class imbalance and sample representativeness issues. In Sect. 4, we report an experimental study on controlling the class balance of the training and test sets; the results are discussed critically and comparatively. In Sect. 5, we highlight the contributions of this paper and provide further directions towards dealing effectively with the issue of sample representativeness, as well as how to use our framework to change the class balance in the training set for highly imbalanced data sets to further address poor performance due to class imbalance.

2 Theoretical preliminaries

In this section, we provide theoretical preliminaries on data partitioning and granular computing. In particular, we describe two ways of machine learning experimentation through data partitioning, namely cross-validation and partitioning into training/test sets. In addition, we describe the concepts of information granules and information granularity which are used in the proposed framework described in Sect. 3.

2.1 Data partitioning

In machine learning, there are several ways of data partitioning for experimentation. The most popular ways are typically referred to as training/test partitioning or cross-validation (Kohavi 1995; Geisser 1993; Devijver 1982).

The training/test partitioning typically involves the partitioning of the data into a training set and a test set in a specific ratio, e.g., 70% of the data are used as the training set and 30% of the data are used as the test set. This data partitioning can be done randomly or in a fixed way (e.g. the first 70% of the instances in the data set are assigned to training set and the rest to the test set). The fixed way is typically avoided (except when order matters) as it may introduce systematic differences between the training set and the test set, which leads to sample representativeness issues. To avoid such systematic differences, the random assignment of instances into training and test sets is typically used.

Cross-validation is conducted by partitioning a data set into n folds (or subsets), followed by an iterative process of combining the folds into different training and test sets. For n folds, there will be n iterations, where at each iteration, one of the folds is used as the test set, while the others, i.e., $n-1$ folds, are used as the training set. In other words,

each of the n folds is, in turn, used as the test set at one of the n iterations, while the rest of the folds are combined together as the training set. In laboratory research, tenfold cross-validation is a popular practice, i.e., the original data set is partitioned into ten subsets. Cross-validation is generally more expensive in terms of computational cost than training/test partitioning.

There have been some new perspectives identified in Liu et al. (2017) regarding the two above ways of data partitioning used for machine learning experimentation. In particular, cross-validation is considered as an effective measure of the learnability of an algorithm, i.e., the degree to which the algorithm is suitable to learn a high-quality model from the given training data. This is to enable appropriate employment of the suitable learning algorithms towards producing predictive models on the basis of existing data. The way of partitioning a data set into a training set and a test set is taken typically towards learning a model that covers highly complete patterns from the training data and evaluating the model accuracy using highly similar but different instances from the test data. This is to make sure that the model accuracy is evaluated in a trustworthy way using a suitable test set. Section 3 will present a proposed approach for more effective partitioning of data into a training set and a test set.

2.2 Granular computing

Granular computing has been an increasingly popular approach for in-depth processing of information. It is aimed at structural thinking at the philosophical level, as well as at structural problem solving at the practical level (Yao 2005). In general, granular computing involves two operations, namely, granulation and organization. The former operation means to decompose a whole into several parts, whereas the latter operation means to integrate several parts into a whole. From computer science perspective, granulation corresponds to the top-down approach and organization corresponds to the bottom-up approach. The nature of granular computing involves two commonly used concepts, namely, granule and granularity.

In the context of information granule, a granule is defined as "a small particle, especially, one of numerous particles forming a larger unit", according to the Merriam-Webster Dictionary (Merriam-Webster 2016). In practice, there have been various examples of granules in broad application areas.

In the setting of set theory, a set of any formalism can be viewed as a granule, since a set is a collection of elements. In this context, each element is viewed as a particle. Different formalisms of sets include deterministic sets (Liu et al. 2016), probabilistic sets (Liu et al. 2016), fuzzy sets (Zadeh 2015), and rough sets (Pedrycz 2011).

In the area of computer science, a granule can act as a class due to the fact that a class is a group of objects which are highly similar to each other. An object can also be viewed as a granule, since each object involves a number of attributes, each of which is considered as a particle. Moreover, a granule can also act as a cluster due to the fact that clustering is another way of grouping objects.

In the area of natural languages, a document could be organized in different forms of text units, such as chapters, sections, paragraphs, sentences, and words. In this context, each form of text units can be viewed as a special type of granule. Moreover, each word is viewed as the finest granule due to the fact that a word consists of letters, each of which is viewed as a particle (Liu and Cocea 2017b).

The concept of information granules is also popularly involved in other application areas, such as image processing, machine learning, and rule-based systems. More details on information granules can be found in Pedrycz (2011), and Pedrycz and Chen (2011, 2015a, b).

In the context of information granularity, information granules can be located in different levels of granularity. In set theory, a set S may have several subsets $(S_1, S_2, \ldots, S_n)$ and each subset may also have several subsubsets $(S_{1.1}, S_{1.2}, \ldots, S_{1.m}, \ldots, S_{n.1}, S_{n.2}, \ldots, S_{n.m})$. In this context, the set S is a granule in the top level of granularity, the subsets $(S_1, S_2, \ldots, S_n)$ are in the middle level of granularity, and the subsubsets $(S_{1.1}, S_{1.1}, \ldots, S_{1.m}, \ldots, S_{n.1}, S_{n.2}, \ldots, S_{n.m})$ are in the bottom level of granularity. In computer science, a class can be specialized into several subclasses through information granulation. In addition, subclasses can be generalized into a super class through information organization.

In natural language processing, a document can be managed in a granular structure, as illustrated in Fig. 1. In particular, the complexity of a text instance (granule) can be reduced through top-down decomposition (granulation) to enable text units (granules) in different levels of granularity (such as paragraphs, sentences, and words) to be processed separately. In addition, the outcomes for processing text units in the same level of granularity can be combined through bottom-up aggregation (organization) towards deriving the outcome for processing larger text units in a higher level of granularity.

In real applications, techniques of granular computing have been involved very often in other popular areas, such as artificial intelligence (Wilke and Portmann 2016; Yao 2005; Skowron et al. 2016), computational intelligence (Dubois and Prade 2016; Yao 2005; Kreinovich 2016; Livi and Sadeghian 2016), and machine learning (Min and Xu 2016; Peters and Weber 2016; Liu and Cocea 2017a; Antonelli et al. 2016).

Furthermore, ensemble learning is also a subject that involves applications of granular computing concepts (Liu

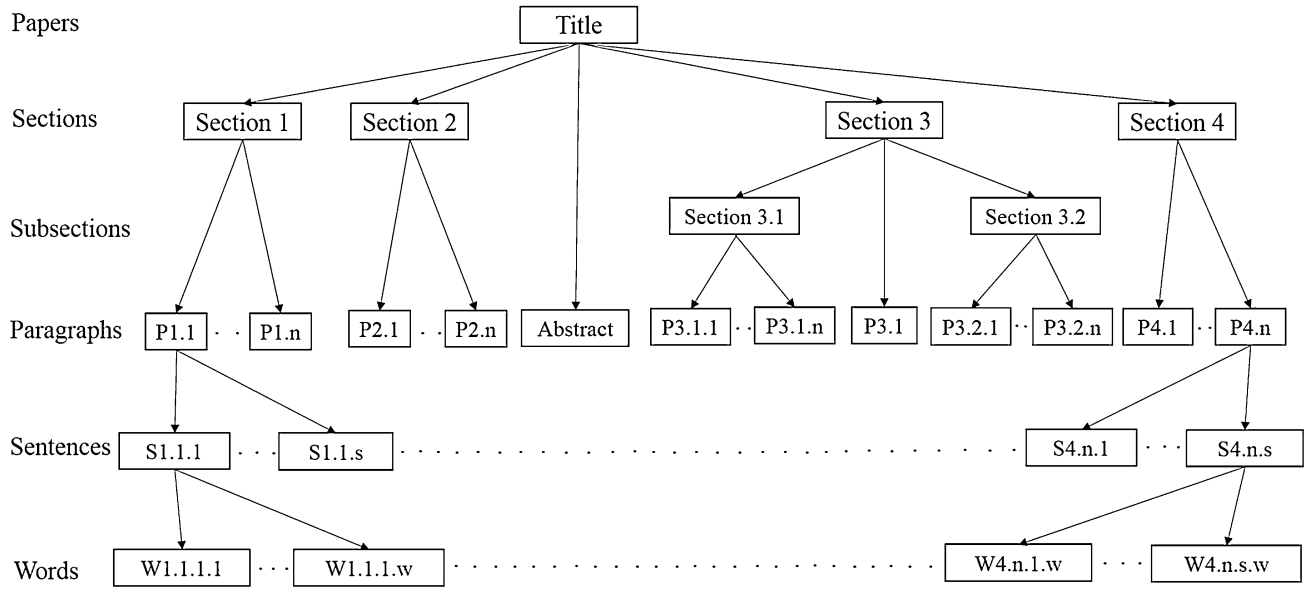

Fig. 1 Fuzzy information granulation for text processing (Liu and Cocea 2017b)

and Cocea 2017a). In particular, ensemble learning approaches, such as Bagging, involve information granulation through decomposing a training set into a number of overlapping samples and combining the predictions made from different classifiers towards classifying a test instance; a similar perspective has also been stressed and discussed in Hu and Shi (2009). Section 3 will show how granular computing concepts can be used towards more effective partitioning of data for machine learning experimentation.

3 Semi-random partitioning of data into training and test sets

In this section, we propose a multi-granularity framework for effective control of the partitioning of a data set into a training set and a test set. We also justify how the proposed approach can address the class imbalance and sample representativeness issues that can arise from random partitioning.

3.1 Key features

The multi-granularity framework for semi-random data partitioning is illustrated in Fig. 2. In particular, this framework involves three levels of granularity as outlined below:

1. *Level 1* Data Partitioning is done randomly on the basis of the original data set towards getting a training set and a test set.
2. *Level 2* The original data set is divided into a number of subsets, with each subset containing a class of instances. Within each subset (i.e., all instances with a particular class label), data partitioning into training and test sets is done randomly. The training and test sets for the whole data set are obtained by merging all the training and test subsets, respectively.
3. *Level 3* Based on the subsets obtained in Level 2, each of them is divided again into a number of subsubsets, where each of the subsubsets contains a subclass (of the corresponding class) of instances. The data partitioning is done randomly within each subsubset. The training and test sets for the whole data set are obtained by merging all the training and test subsubsets, respectively.

In this multi-granularity framework, Level 2 is aimed at addressing the class imbalance issue, i.e., to control the distribution of instances by class *within* the training and test sets. Level 3 is aimed at addressing the issue of sample representativeness, i.e., it is to avoid the case that the training instances are highly dissimilar to the test instances following the data partitioning.

In the setting of granular computing, the proposed framework involves explicitly both granulation and organization. In particular, granulation is involved through the operation that a data set is divided into a number of subsets and each subset is divided into a training subset and a test subset (Level 2), or further divided into subsubsets and then split into training and test subsubsets (Level 3). In addition, organization is involved by integrating the training subsets or subsubsets into a whole training set, and the

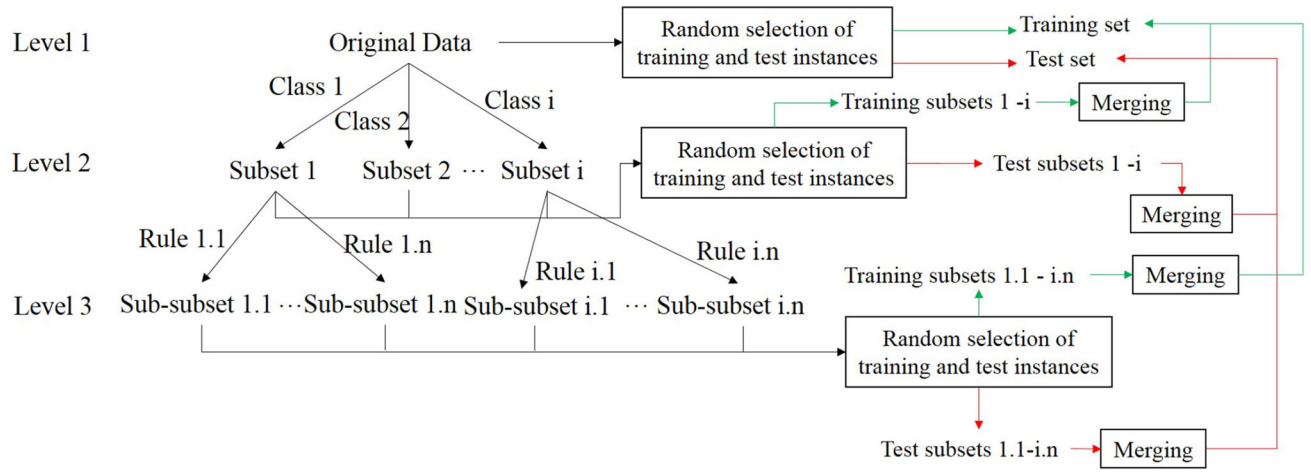

Fig. 2 Multi-granularity framework for semi-random data partitioning

test subsets or subsubsets into a whole test set. In addition, in each level of the granularity as shown in Fig. 2, a set of data is viewed as a granule, which also has hierarchical relationships with sets of data (granules) located in other levels of granularity.

3.2 Justification

Level 2 of the proposed multi-granularity framework is aimed at controlling effectively the selection of training/test instances towards avoiding the issue of class imbalance, especially when the original data set is balanced. In particular, Level 2 is designed to ensure that for each class of instances, a fixed percentage of the instances would be included in the training/test set. For example, if we suppose that a data set is divided into a training set and a test set in the ratio of 70:30, the strategy of semi-random data partitioning involved in Level 2 of the multi-granularity framework can ensure that for each class of instances, there would be 70% of the instances selected as training instances and the rest of them selected as test instances. The above statement can be proven as follows:

Let us suppose that a data set contains two classes (positive and negative) of instances with the frequency distribution of $p : (1-p)$, and the size of the data set is m. Following data partitioning, the percentage of the training set is q, whereas the percentage of the test set is $1-q$.

While the above strategy of semi-random data partitioning is taken, the following steps would be involved:

1. *Step 1* The data set is divided into two subsets, respectively, for the positive and negative classes, which results in mp positive instances and $m(1-p)$ negative instances.
2. *Step 2* Each class subset is partitioned into a training subset and a test subset. In particular, for the positive class, the size of the training subset is mpq and the size of the test subset is $mp(1-q)$. Similarly, for the negative class, the size of the training subset is $m(1-p)q$ and the size of the test subset is $m(1-p)(1-q)$.
3. *Step 3* The two training subsets resulting from Step 2 are merged into a whole training set and the frequency distribution between the positive and negative classes is $mpq : m(1-p)q$, which is equivalent to $p : (1-p)$, i.e., the original class distribution.
4. *Step 4* The two test subsets resulting from Step 2 are merged into a whole test set and the frequency distribution between the positive and negative classes is $mp(1-q) : m(1-p)(1-q)$, which is equivalent to $p : (1-p)$, i.e., the original class distribution.

Thus, the procedure for Level 2 ensures that the original class distribution for the whole data set is reflected within the training and test sets. The above proof, although demonstrated for a 2-class problem, also applies to multi-class classification problems, since the frequency distribution between different classes does not have any dependency on the number of classes as shown above.

The above procedure is inspired from the stratified sampling technique, used in statistics (Srndal et al. 1992). In this context, a population (data set) is divided into subpopulations (data subsets), and then, simple random sampling is used within each subpopulation for getting a subsample (strata). In the context of machine learning, each class represents a subpopulation and a training/test subset for a class represents a strata. Stratified sampling is typically used for improving the sample representiveness by reducing the data variability and thus reducing sampling error (Esfahani and Dougherty 2014; Lang et al. 2016).

However, for the purpose of avoiding class imbalance through preserving the class distributions for training and test sets, the classic stratified sampling technique needs to calculate the size of each strata based on its percentage of the total, whereas the procedure for Level 2 of the proposed

Table 1 Sampling probability by stratified sampling

Weight	Probability for class 1 (%)	Probability for class 2 (%)	Probability for class 3 (%)
Training set: 70%	28	28	14
Test set: 30%	12	12	6

multi-granularity framework only needs to divide a data set into subsets (each subset for a class) and then partition (in a fixed ratio) each subset into a training subset and a test subset, without the need to calculate the size of each training/test subset.

For example, a data set has three classes with the distribution 40:40:20; the data partitioning needs to result in 70% of the data set for the training subset and 30% for the test subset.

While stratified sampling is adopted, Table 1 shows that each class needs to be given a probability for its instances to be selected into either the training set or the test set, i.e., it is needed to calculate the sampling probability for each class regarding the selection of its instances for the training/test set. This way aims to preserve the original class distribution in both the training and test sets but leads to higher computational complexity.

Table 2 shows that it is not needed to calculate the sampling probability for each class regarding the selection of its instances for the training/test set. Instead, it is only needed to divide the original data set into n subsets, where n is the number of classes. For each subset corresponding to a class, it is just simply selecting an instance for the training/test set with 70%/30% chance.

On the basis of the above description, stratified sampling pays only attention to preserving the original class distribution by giving each class a sampling probability for its instances to be selected, without taking into account the balance between training and test samples, whereas the proposed semi-random partitioning pays more attention to balancing training and test sets by simply giving each instance 70%/30% chance to be selected for the training/test set.

Level 3 of the proposed multi-granularity framework is aimed at controlling effectively the selection of training/test instances to ensure sample representativeness. In particular, the lack of sample representativess is likely to lead to overfitting, which means that a model performs well on the training data, but poorly on the test data. Thus, what the algorithm learns from the training data is not useful for the test data—something that is typically referred as a lack of generalization; in other words, the model is too specialized, i.e., it has learned from the training data very well, but cannot generalize this knowledge to other situations such as the ones in the test set.

To avoid this problem, the sample of data in the training set should be representative of the whole data, by ensuring that there is not a large dissimilarity between the training set and the test set. To avoid this dissimilarity, level 3 of the proposed multi-granularity framework is thus designed to involve grouping instances on the basis of their similarity to each other, and perform the partitioning within these groups, such that instances from the group will be present in both the training and the test sets.

4 Experiments, results, and discussion

In this section, we report two experimental studies. In particular, the first study involves comparing our proposed approach of semi-random data partitioning with the stratified sampling approach. The second study is to validate the effectiveness of the strategy of semi-random data partitioning involved in Level 2 of the multi-granularity framework proposed in Sect. 3. In particular, we compare the strategy of the semi-random data partitioning with the one of the traditional random data partitioning, in terms of class frequency distribution within the training and test sets, as well as the influence of this distribution on classification performance.

The experimental studies are conducted using 12 UCI data sets (Lichman 2013). The characteristics of the data sets are shown in Table 3. All the chosen data sets are either balanced or slightly imbalanced, except for the 'anneal' and 'autos' data sets, in terms of class frequency distribution. For using both balanced or slightly imbalanced data sets, the aim is to show that it is necessary to manage to keep the balance level of both the training and test sets as close to the balance level of the original data set as possible, towards avoiding any impact on the learning performance of the algorithms and on the classification

Table 2 Sampling probability by semi-random partitioning

Weight	Probability for class 1 (%)	Probability for class 2 (%)	Probability for class 3 (%)
Training set: 70%	70	70	70
Test set: 30%	30	30	30

Table 3 Data sets

Data set	Feature types	#Attributes	#Instances	#Classes
Anneal	Discrete, continuous	38	798	6
Autos	Discrete, continuous	26	205	7
Credit-a	Discrete, continuous	15	690	2
Heart-stalog	Continuous	13	270	2
Iris	Continuous	4	150	3
kr-vs-kp	Discrete	36	3196	2
Labor	Discrete, continuous	17	57	2
Segment	Continuous	19	2310	7
Sonar	Continuous	60	208	2
Tae	Discrete, continuous	6	151	3
Vote	Discrete	16	435	2
Wine	Continuous	13	178	3

performance of the learned classifiers. The imbalanced data sets, i.e., 'anneal' and 'autos', as well as the 'segment' balanced data set, have a larger number of classes, while the other nine data sets have two or three classes. These will allow us to analyze the results in terms of number of classes, as well.

Three popular machine learning algorithms, i.e., the C4.5 decision tree learning algorithm (Quinlan 1993), Naive Bayes (Rish 2001), and K-nearest neighbour (Liu et al. 2016), are used for validation, since these three algorithms are all sensitive to class imbalance (Longadge et al. 2013).

Regarding the first experimental study, the results are shown in Tables 4, 5, and 6. In these three tables, SS stands for stratified sampling and SR stands for semi-random partitioning.

Table 4 shows that the proposed semi-random partitioning outperforms stratified sampling in 9 out of 12 cases, and the two approaches perform the same in the other 3 cases, in terms of overall accuracy of classification. In addition, the proposed semi-random partitioning outperforms stratified sampling in terms of precision and recall with respect to each single class in most cases.

Table 5 shows that the proposed semi-random partitioning outperforms stratified sampling in 9 out of 12 cases, and the two approaches perform the same in 2 out of the other 3 cases, in terms of overall accuracy of classification. In addition, the proposed semi-random partitioning outperforms stratified sampling in terms of precision and recall with respect to each single class in most cases.

Table 6 shows that the proposed semi-random partitioning outperforms stratified sampling in 7 out of 12 cases, and the two approaches perform the same in 3 out of the other 5 cases, in terms of overall accuracy of classification. In addition, the proposed semi-random partitioning outperforms stratified sampling in terms of precision and recall with respect to each single class in most cases.

Regarding the second experimental study, Table 7 displays the original distribution of instances across classes for each data set in terms of frequency (designated by #) and percentages (designated by %). For example, the anneal data set (first row in Table 7) has 6 classes, and in the original distribution, class 1 has 8 instances (representing 1% of all instances), class 2 has 99 instances (representing 11% of the data), and so on. The same information is also displayed for the training and test sets used with the semi-random partitioning approach. The percentage numbers have been rounded to integers for ease of comparison. The loss of precision due to this rounding means that the sum across all classes may not be precisely 100%. In addition, when the number of instances is low, a small difference in the number of instances may lead to a much bigger difference in the percentages values.

Tables 8, 9, 10 show the original distribution, as well as the distribution within the training and test sets for C4.5, NB, and K-NN, respectively. The original distribution was included in all tables for ease of comparison.

The random selection of data for training and test sets leads to different effects on the distribution of instances across classes within the training and test sets, which are outlined below:

- For initially balanced data sets such as 'iris', 'segment', and 'tae', the random partitioning may lead to a loss of balance within the training and test sets; this loss can be observed for C4.5 on the 'iris' and 'tae' data sets, while for the 'segment' data set, the variation is smaller; similarly, for NB, the loss of balance can be noticed for the 'iris' and 'tae' data sets, while for the 'segment' data set, the variation is smaller, but more noticeable than for C4.5; for K-NN, a loss of balance can be observed for the 'tae' data set, while for the iris data set, the imbalance is very small, and for the 'segment'

Table 4 Comparison with stratified sampling in terms of C4.5 performance

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Anneal								
SS								
Precision	0.00	0.88	0.99	0.00	1.00	1.00		0.98
Recall	0.00	1.00	0.98	0.00	1.00	1.00		
SR								
Precision	0.00	0.97	0.99	0.00	1.00	1.00		0.99
Recall	0.00	1.00	1.00	0.00	1.00	1.00		
Autos								
SS								
Precision	0.00	0.00	0.63	0.88	0.72	0.80	0.80	0.77
Recall	0.00	0.00	0.71	0.7	0.81	0.80	1.00	
SR								
Precision	0.00	0.50	1.00	0.95	0.69	0.55	0.89	0.79
Recall	0.00	1.00	0.57	0.95	0.69	0.60	1.00	
Credit-a								
SS								
Precision	0.80	0.85						0.83
Recall	0.82	0.83						
SR								
Precision	0.82	0.97						0.89
Recall	0.97	0.83						
Heart-statlog								
SS								
Precision	0.80	0.68						0.74
Recall	0.71	0.78						
SR								
Precision	0.79	0.89						0.83
Recall	0.93	0.69						
Iris								
SS								
Precision	1.00	0.93	0.93					0.96
Recall	1.00	0.93	0.93					
SR								
Precision	1.00	1.00	0.94					0.98
Recall	1.00	0.93	1.00					
kr-vs-kp								
SS								
Precision	0.99	1.00						0.99
Recall	1.00	0.99						
SR								
Precision	0.99	1.00						0.99
Recall	1.00	0.99						
Labor								
SS								
Precision	0.80	0.85						0.83
Recall	0.67	0.92						
SR								
Precision	0.83	0.91						0.88
Recall	0.83	0.91						

Table 4 continued

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Segment								
SS								
Precision	0.98	1.00	0.89	0.92	0.84	0.99	1.00	0.95
Recall	0.98	1.00	0.90	0.92	0.83	1.00	0.99	
SR								
Precision	0.97	1.00	0.89	0.99	0.88	1.00	1.00	0.96
Recall	0.97	1.00	0.89	0.94	0.93	1.00	1.00	
Sonar								
SS								
Precision	0.65	0.72						0.68
Recall	0.69	0.68						
SR								
Precision	0.81	0.87						0.84
Recall	0.86	0.82						
Tae								
SS								
Precision	0.40	0.39	0.46					0.41
Recall	0.53	0.33	0.38					
SR								
Precision	0.55	0.67	0.55					0.57
Recall	0.73	0.27	0.69					
Vote								
SS								
Precision	0.94	0.98						0.96
Recall	0.96	0.96						
SR								
Precision	0.97	0.94						0.96
Recall	0.96	0.96						
Wine								
SS								
Precision	1.00	0.96	0.93					0.96
Recall	1.00	0.96	0.93					
SR								
Precision	1.00	0.91	1.00					0.96
Recall	0.94	1.00	0.93					

data set, the variation is small and similar to the variation for C4.5.

- For slightly imbalanced data sets, the random partitioning may lead to a more balanced distribution in the training set, but a more imbalanced one in the test set, i.e., for C4.5., 'heart-statlog'; for NB, labor, and vote; for K-NN, 'credit-a', 'labor', and 'sonar'. Sometimes, the imbalance in the test set may mean that the majority class from the training set becomes minority class in the test set— this occurs only for one data set, i.e., 'sonar' with K-NN, which is probably due to the fact that the distribution in this data set is very close to perfect balance (47:53).
- For slightly imbalanced data sets, the random partitioning may lead to a more balanced distribution in the test set, but a more imbalanced distribution in the training set, i.e., for C4.5, 'kr-vs-kp', and 'labor' by C4.5; for NB, 'heart-statlog'. For two of these, C4.5—'kr-vs-kp' and NB—'heart-statlog', in the test set, the majority class is reversed in comparison with the training set.
- For slightly imbalanced data sets, the random partitioning may lead to both the training and test sets to become more imbalanced, with a different class being the majority class in the training and test sets; for example, in the 'sonar' data set with C4.5, class 2 is the

Table 5 Comparison with stratified sampling in terms of NB performance

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Anneal								
SS								
Precision	1.00	0.87	0.98	0.00	1.00	1.00		0.93
Recall	1.00	0.87	0.92	0.00	1.00	0.50		
SR								
Precision	0.50	0.79	0.99	0.00	1.00	0.30		0.86
Recall	1.00	1.00	0.82	0.00	1.00	0.92		
Autos								
SS								
Precision	0.00	0.00	1.00	0.46	0.65	0.44	0.50	0.53
Recall	0.00	0.00	0.14	0.6	0.81	0.40	0.38	
SR								
Precision	0.00	1.00	0.42	0.80	0.55	0.20	0.67	0.53
Recall	0.00	1.00	0.71	0.40	0.69	0.20	0.75	
Credit-a								
SS								
Precision	0.77	0.87						0.82
Recall	0.85	0.79						
SR								
Precision	0.91	0.78						0.83
Recall	0.67	0.95						
Heart-statlog								
SS								
Precision	0.91	0.82						0.86
Recall	0.84	0.89						
SR								
Precision	0.86	0.94						0.89
Recall	0.96	0.81						
Iris								
SS								
Precision	1.00	0.93	0.88					0.93
Recall	1.00	0.87	0.93					
SR								
Precision	1.00	1.00	1.00					1.00
Recall	1.00	1.00	1.00					
kr-vs-kp								
SS								
Precision	0.86	0.89						0.88
Recall	0.91	0.84						
SR								
Precision	0.88	0.89						0.89
Recall	0.91	0.87						
Labor								
SS								
Precision	1.00	0.86						0.89
Recall	0.67	1.00						
SR								
Precision	1.00	1.00						1.00
Recall	1.00	1.00						

Table 5 conitinued

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Segment								
SS								
Precision	1.00	1.00	0.68	0.53	0.49	1.00	1.00	0.75
Recall	0.48	1.00	0.87	0.85	0.56	0.51	0.99	
SR								
Precision	0.79	1.00	0.57	0.90	0.43	0.95	1.00	0.80
Recall	0.97	1.00	0.12	0.87	0.68	0.97	1.00	
Sonar								
SS								
Precision	0.92	0.66						0.71
Recall	0.41	0.97						
SR								
Precision	0.73	0.83						0.77
Recall	0.83	0.73						
Tae								
SS								
Precision	0.41	0.44	0.46					0.44
Recall	0.47	0.53	0.31					
SR								
Precision	0.65	0.63	0.69					0.65
Recall	0.73	0.67	0.56					
Vote								
SS								
Precision	0.84	0.96						0.91
Recall	0.94	0.89						
SR								
Precision	0.97	0.83						0.91
Recall	0.88	0.96						
Wine								
SS								
Precision	1.00	0.92	1.00					0.96
Recall	0.94	1.00	0.93					
SR								
Precision	0.94	0.95	1.00					0.98
Recall	0.97	1.00	1.00					

majority class in the training set, while class 1 is the majority class in the test set. This situation occurs on the 'sonar' data set for C4.5 and NB, and on the 'wine' data set for all algorithms (C4.5, NB, and K-NN).

- For the data sets with a high number of classes and an imbalanced distribution, e.g., anneal and autos, the random partitioning may preserve the original distribution for some classes, while for others, there is an imbalance in the training set, the test set or both, i.e., the 'autos' data set for all algorithms (C4.5, NB, and K-NN); sometimes, the majority class in the training set is no longer the majority class in the test set, e.g., for C4.5—'autos', class 5 is the majority class in the training set, while class 4 is the majority class in the test set (as well as the original data set). For the anneal data set, the distribution changes slightly, but the majority of the changes are less than 2%—for this reason, we consider that the distribution for this data set with all algorithms is very similar to the original distribution.
- For all data sets, the random partitioning may lead to a very similar distribution in the training and test sets as in the original data set. i.e., for C4.5, 'anneal', 'credit-a', and 'vote'; for NB, 'anneal', 'credit-a', and 'kr-vs-kp'; for K-NN, 'anneal', 'heart-statlog', 'kr-vs-kp', and 'vote'.

Table 11 shows the experimental results for the C4.5 algorithm with random (R) and semi-random (SR)

Table 6 Comparison with stratified sampling in terms of K-NN performance

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Anneal								
SS								
Precision	0.00	0.63	0.86	0.00	0.75	0.83		0.83
Recall	0.50	1.00	0.98	0.00	1.00	0.62		
SR								
Precision	1.00	0.90	0.99	0.00	1.00	0.69		0.96
Recall	1.00	0.93	0.96	0.00	1.00	0.92		
Autos								
SS								
Precision	0.00	0.00	0.00	0.33	0.54	0.11	0.00	0.32
Recall	0.00	0.00	0.00	0.60	0.44	0.10	0.00	
SR								
Precision	0.00	0.00	0.71	0.58	0.55	0.50	0.67	0.58
Recall	0.00	0.00	0.71	0.55	0.75	0.40	0.50	
Credit-a								
SS								
Precision	0.66	0.71						0.69
Recall	0.63	0.74						
SR								
Precision	0.91	0.88						0.89
Recall	0.84	0.93						
Heart-statlog								
SS								
Precision	0.64	0.54						0.59
Recall	0.60	0.58						
SR								
Precision	0.84	0.88						0.85
Recall	0.91	0.79						
Iris								
SS								
Precision	1.00	1.00	0.94					0.98
Recall	1.00	0.88	0.92					
SR								
Precision	1.00	0.88	1.00					0.96
Recall	1.00	1.00	0.87					
kr-vs-kp								
SS								
Precision	0.52	0.00						0.52
Recall	1.00	0.00						
SR								
Precision	0.94	0.97						0.96
Recall	0.97	0.94						
Labor								
SS								
Precision	0.86	1.00						0.94
Recall	1.00	0.92						
SR								
Precision	1.00	0.92						0.94
Recall	0.83	1.00						

Table 6 continued

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Segment								
SS								
Precision	0.96	1.00	0.88	0.93	0.89	1.00	1.00	0.95
Recall	0.96	1.00	0.92	0.89	0.88	1.00	1.00	
SR								
Precision	0.96	1.00	0.85	0.99	0.87	0.96	1.00	0.95
Recall	0.98	1.00	0.95	0.86	0.83	1.00	1.00	
Sonar								
SS								
Precision	0.88	0.82						0.84
Recall	0.76	0.91						
SR								
Precision	0.84	0.78						0.81
Recall	0.72	0.88						
Tae								
SS								
Precision	0.25	0.43	0.40					0.37
Recall	0.20	0.40	0.50					
SR								
Precision	0.54	0.57	0.63					0.59
Recall	0.47	0.53	0.75					
Vote								
SS								
Precision	0.89	0.97						0.94
Recall	0.96	0.93						
SR								
Precision	0.96	0.90						0.94
Recall	0.94	0.94						
Wine								
SS								
Precision	0.84	0.65	0.44					0.65
Recall	0.94	0.50	0.53					
SR								
Precision	0.95	1.00	0.93					0.96
Recall	1.00	0.90	1.00					

partitioning, which include the accuracy (last column), as well as precision and recall per class.

In terms of accuracy, the results show three situations: (a) the semi-random partitioning leads to the same accuracy as random partitioning, i.e., 'anneal', 'kr-vs-kp', and 'segment'; (b) the semi-random partitioning displays small improvements in accuracy (up to 3%), i.e., 'autos', 'credit-a', 'heart-statlog', 'sonar', 'tae', and 'vote'; (c) the semi-random partitioning displays large improvements in accuracy (5% or more), i.e., iris (7%), labor (23%), and wine (5%).

Figures 3, 4, 5 display the class distribution, as well as the precision and recall for the experiments with C4.5 on all data sets (4 per graph). The class distribution for the whole data set is represented by the middle bar for every class; the distribution into the training and test sets for random partitioning is represented by the bars on the left, while the ones for semi-random partitioning are represented by the bars on the right. The lines with the square points represent the values for precision—yellow for random partitioning and brown for semi-random partitioning; the lines with the triangle points represent the values for recall—blue for random partitioning and green for semi-random partitioning. The left axis on the graphs represents the number of instances (or class frequency), while the right axis represents the values for precision and recall, with a range from 0 to 1.

Table 7 Class frequency distribution with semi-random partitioning

Data set	Original distribution	Training set	Test set
Anneal			
#	8:99:684:0:67:40	6:69:479:0:47:28	2:30:205:0:20:12
%	1:11:76:0:7:4	1:11:76:0:7:4	1:11:76:0:7:4
Autos			
#	0:3:22:67:54:32:27	0:2:15:47:38:22:19	0:1:7:20:16:10:8
%	0:1:11:33:26:16:13	0:1:10:33:27:15:13	0:2:11:32:26:16:13
Credit-a			
#	307:383	215:268	92:115
%	44:56	45:55	44:56
Heart-statlog			
#	150:120	105:84	45:36
%	56:44	56:44	56:44
Iris			
#	50:50:50	35:35:35	15:15:15
%	33:33:33	33:33:33	33:33:33
kr-vs-kp			
#	1669:1527	1168:1069	501:458
%	52:48	52:48	52:48
Labor			
#	20:37	14:26	6:11
%	35:65	35:65	35:65
Segment			
#	330:330:330:330:330:330:330	231:231:231:231:231:231:231	99:99:99:99:99:99:99
%	14:14:14:14:14:14:14	14:14:14:14:14:14:14	14:14:14:14:14:14:14
Sonar			
#	97:111	68:78	29:33
%	47:53	47:53	47:53
Tae			
#	49:50:52	34:35:36	15:15:16
%	32:33:34	32:33:34	33:33:35
Vote			
#	267:168	187:118	80:50
%	61:39	61:39	62:38
Wine			
#	59:71:48	41:50:34	18:21:14
%	33:40:27	33:40:27	34:40:26

For the data sets where the accuracy is the same for both random and semi-random partitioning, i.e., 'anneal', 'kr-vs-kp', and 'segment', the class distribution (see Table 8; Figs. 3, 4) is very similar for both random and semi-random partitioning. For the 'kr-vs-kp' data set, although the test set is more balanced (and the training one more imbalanced) compared with the original distribution, the change is very small, especially for the training set where the change is of 1%. For this data set, we also observed that the majority class in the training set becomes the minority class in the test set—the difference, however, is very small, i.e., 2%. Given the large size of this data set and only a slight imbalance in the distribution of classes, it is not surprising that such a small change in distribution does not impact the results.

For the data sets where the accuracy is slightly higher when semi-random partitioning is used, i.e., 'autos', 'credit-a', 'heart-statlog', 'sonar', 'tae', and 'vote', the random partitioning has different effects on the class distribution within the training and test sets.

For the 'autos' data set, we notice several situations for different classes:

(a) for class 2, all instances are assigned to the training set; thus, while the model learned something about

Table 8 C4.5: class frequency distribution in training and test sets for random partitioning

Data set	Original distribution	Training set	Test set
Anneal			
#	8:99:684:0:67:40	7:73:483:0:39:27	1:26:201:0:28:13
%	1:11:76:0:7:4	1:12:77:0:6:4	0:10:75:0:10:5
Autos			
#	0:3:22:67:54:32:27	0:3:17:41:43:23:17	0:0:5:26:11:9:10
%	0:1:11:33:26:16:13	0:2:12:28:30:16:12	0:0:8:43:18:15:16
Credit-a			
#	307:383	211:272	96:111
%	44:56	44:56	46:54
Heart-statlog			
#	150:120	99:90	51:30
%	56:44	52:48	63:37
Iris			
#	50:50:50	38:30:37	12:20:13
%	33:33:33	36:29:35	27:44:29
kr-vs-kp			
#	1669:1527	1196:1041	473:486
%	52:48	53:47	49:51
Labor			
#	20:37	13:27	7:10
%	35:65	33:68	41:59
Segment			
#	330:330:330:330:330:330:330	223:223:230:239:242:229:231	107:107:100:91:88:101:99
%	14:14:14:14:14:14:14	14:14:14:15:15:14:14	15:15:14:13:13:15:14
Sonar			
#	97:111	62:84	35:27
%	47:53	42:58	56:44
Tae			
#	49:50:52	34:32:40	15:18:12
%	32:33:34	32:30:38	33:40:27
Vote			
#	267:168	186:119	81:49
%	61:39	61:39	62:38
Wine			
#	59:71:48	42:55:28	17:16:20
%	33:40:27	34:44:22	32:30:38

this class, nothing is tested and, consequently, the performance for this class is 0;

(b) for class 3 and class 6, the random distribution leads to proportionately more instances in the training set for random partitioning than for the semi-random one—for these, the performance is higher with the random partitioning, which could be explained by the more opportunities for learning for the random partitioning and/or by the lack of sample representativeness for the semi-random partitioning; this will be discussed in more detail further on;

(c) for class 4 and class 7, the opposite situation occurs, i.e., for the random partitioning, there are proportionally less instances in the training set for random partitioning than for the semi-random one—for these, the performance is higher with the semi-random partitioning; similarly, this could be due to lack of learning opportunities for the random partitioning and/or sample representativeness for the semi-random partitioning;

(d) finally, for class 5, there are proportionally more instances in the training set for random partitioning

Table 9 NB: class frequency distribution in training and test sets for random partitioning

Data set	Original distribution	Training set	Test set
Anneal			
#	8:99:684:0:67:40	4:67:484:0:44:30	4:32:200:0:23:10
%	1:11:76:0:7:4	1:11:77:0:7:5	1:12:74:0:9:4
Autos			
#	0:3:22:67:54:32:27	0:2:15:45:39:23:20	0:1:7:22:15:9:7
%	0:1:11:33:26:16:13	0:1:10:31:27:16:14	0:2:11:36:25:15:11
Credit-a			
#	307:383	216:267	91:116
%	44:56	45:55	44:56
Heart-statlog			
#	150:120	111:78	39:42
%	56:44	59:41	48:52
Iris			
#	50:50:50	37:31:37	13:19:13
%	33:33:33	35:30:35	29:42:29
kr-vs-kp			
#	1669:1527	1164:1073	505:454
%	52:48	52:48	53:47
Labor			
#	20:37	16:24	4:13
%	35:65	40:60	24:76
Segment			
#	330:330:330:330:330:330:330	245:228:229:220:245:218:232	85:102:101:110:85:112:98
%	14:14:14:14:14:14:14	15:14:14:14:15:13:14	12:15:15:16:12:16:14
Sonar			
#	97:111	60:86	37:25
%	47:53	41:59	60:40
Tae			
#	49:50:52	34:31:41	15:19:11
%	32:33:34	32:29:39	33:42:24
Vote			
#	267:168	183:122	84:46
%	61:39	60:40	65:35
Wine			
#	59:71:48	48:43:34	11:28:14
%	33:40:27	38:34:27	21:53:26

than for the semi-random one; in addition, this is the majority class in the test set (while class 4 is the majority one in the training set); for this class, the precision value is higher with semi-random partitioning, while recall is higher with the random partitioning; precision is about how many of the instances labeled by the model are truly class 5 (as opposed to other classes), while recall is about how many of all of the class 5 instances are correctly identified as class 5; thus, a small precision indicates that class 5 instances are wrongly labeled with another class, while a small recall indicates that the model has not learned sufficiently how to identify class 5 (due to either not enough opportunities for learning or due to overfitting); a possible explanation for the higher recall with random partitioning is that the higher number of instances in the training set for the random partitioning leads to a model that has learned "better" how to recognize a class 5 instance based on the knowledge about class 5, while the opposite effect occurs for the semi-random partitioning; the better precision for semi-random

Table 10 K-NN: class frequency distribution in training and test sets for random partitioning

Data set	Original distribution	Training set	Test set
Anneal			
#	8: 99:684:0:67:40	4:64:484:0:50:27	4:35:200:0:17:13
%	1:11:76:0:7:4	1:10:77:0:8:4	1:13:74:0:6:5
Autos			
#	0:3:22:67:54:32:27	0:3:16:49:38:21:17	0:0:6:18:16:11:10
%	0:1:11:33:26:16:13	0:2:11:34:26:15:12	0:0:10:30:26:18:16
Credit-a			
#	307:383	224:259	83:124
%	44:56	46:54	40:60
Heart-statlog			
#	150:120	106:83	44:37
%	56:44	56:44	54:46
Iris			
#	50:50:50	35:34:36	15:16:14
%	33:33:33	33:32:34	33:36:31
kr-vs-kp			
#	1669:1527	1177:1060	492:467
%	52:48	53:47	51:49
Labor			
#	20:37	15:25	5:12
%	35:65	38:63	29:71
Segment			
#	330:330:330:330:330:330:330	220:223:231:238:241:234:230	110:107:99:92:89:96:100
%	14:14:14:14:14:14:14	14:14:14:15:15:14:14	16:15:14:13:13:14:14
Sonar			
#	97:111	64:82	33:29
%	47:53	44:56	53:47
Tae			
#	49:50:52	33:35:38	16:15:14
%	32:33:34	31:33:36	36:33:31
Vote			
#	267:168	186:119	81:49
%	61:39	61:39	62:38
Wine			
#	59:71:48	34:55:36	25:16:12
%	33:40:27	27:44:29	47:30:23

partitioning could be explained by the better balance of distribution between classes with semi-random partitioning, which leads to a model that can distinguish better between a class 5 instance and instances of other classes.

For the 'credit-a' and 'vote' data sets, the class distribution is very similar for random and semi-random partitioning—in this case, the difference is likely to be due to sample representativeness. For the 'heart-statlog', 'sonar', and 'tae', the class distribution changes for the majority of classes when using random partitioning, which has a mixed effect on the results for different classes, i.e., precision and/or recall are sometimes higher for semi-random partitioning and sometimes higher for random partitioning. In addition, when the distribution is similar for random and semi-random partitioning, e.g., class 1 of 'tae' data set, the results are different, which may be due to sample representativeness.

For the data sets where the accuracy is considerably higher when using semi-random partitioning, i.e., iris (7%), labor (23%), and wine (5%), we notice that the random partitioning leads to class distribution imbalance in the training sets, and something in the test sets as well (i.e., iris and wine). The difference in results is likely to be due to

Table 11 C4.5 performance on accuracy, precision, and recall

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Anneal								
R								
Precision	0.00	0.96	0.99	0.00	1.00	1.00		0.99
Recall	0.00	1.00	1.00	0.00	1.00	0.85		
SR								
Precision	0.00	0.97	0.99	0.00	1.00	1.00		0.99
Recall	0.00	1.00	1.00	0.00	1.00	1.00		
Autos								
R								
Precision	0.00	0.00	1.00	0.92	0.50	0.78	0.75	0.77
Recall	0.00	0.00	0.80	0.85	0.73	0.78	0.60	
SR								
Precision	0.00	0.50	1.00	0.95	0.69	0.55	0.89	0.79
Recall	0.00	1.00	0.57	0.95	0.69	0.60	1.00	
Credit-a								
R								
Precision	0.82	0.90						0.86
Recall	0.90	0.83						
SR								
Precision	0.82	0.97						0.89
Recall	0.97	0.83						
Heart-statlog								
R								
Precision	0.97	0.67						0.81
Recall	0.73	0.97						
SR								
Precision	0.79	0.89						0.83
Recall	0.93	0.69						
Iris								
R								
Precision	1.00	0.94	0.81					0.91
Recall	0.92	0.85	1.00					
SR								
Precision	1.00	1.00	0.94					0.98
Recall	1.00	0.93	1.00					
kr-vs-kp								
R								
Precision	0.99	0.99						0.99
Recall	0.99	0.99						
SR								
Precision	0.99	1.00						0.99
Recall	1.00	0.99						
Labor								
R								
Precision	0.67	0.64						0.65
Recall	0.29	0.90						
SR								
Precision	0.83	0.91						0.88
Recall	0.83	0.91						

Table 11 continued

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Segment								
R								
Precision	0.97	0.98	0.92	0.99	0.84	1.00	0.99	0.96
Recall	0.99	1.00	0.91	0.88	0.90	1.00	1.00	
SR								
Precision	0.97	1.00	0.89	0.99	0.88	1.00	1.00	0.96
Recall	0.97	1.00	0.89	0.94	0.93	1.00	1.00	
Sonar								
R								
Precision	0.85	0.79						0.82
Recall	0.83	0.81						
SR								
Precision	0.81	0.87						0.84
Recall	0.86	0.82						
Tae								
R								
Precision	0.50	0.56	0.60					0.56
Recall	0.40	0.56	0.75					
SR								
Precision	0.55	0.67	0.55					0.57
Recall	0.73	0.27	0.69					
Vote								
R								
Precision	0.95	0.96						0.95
Recall	0.98	0.92						
SR								
Precision	0.97	0.94						0.96
Recall	0.96	0.96						
Wine								
R								
Precision	0.94	0.83	0.94					0.91
Recall	0.94	0.94	0.85					
SR								
Precision	1.00	0.91	1.00					0.96
Recall	0.94	1.00	0.93					

the class imbalance issue, as well as sample representativeness (e.g., class 1 of the 'wine' data set has similar distribution for both random and semi-random partitioning, but different precision results).

Table 12 displays the experimental results for Naive Bayes (NB), including recall and precision per class, and accuracy—for random (R) and semi-random (SR) partitioning. Figures 6, 7, 8 display the precision and recall results, as well as the class distribution, with the similar structure as for the previous graphs (with the C4.5 results).

When looking at accuracy, the results for NB show four situations: (a) the semi-random partitioning has lower accuracy than the random one, i.e., 'segment' and 'wine'; (b) the accuracy is the same for both types of partitioning, i.e., 'sonar'; (c) the accuracy for semi-random partitioning is slightly higher than for the random one (up to 4%), i.e., 'anneal', 'autos', 'credit-a', 'iris', 'kr-vskp', and 'vote'; (d) the accuracy for semi-random partitioning is considerably higher (5% or more), i.e., 'heart-statlog'(5%), 'labor' (12%), and 'tae' (18%).

For the data sets displaying lower accuracy for the semi-random partitioning, i.e., 'segment' and 'wine', the difference in accuracy compared with random partitioning is very small, i.e., 1% for 'segment' and 2% for 'wine'. For the 'segment' data set, there is a small change in the class distribution with random partitioning; for the classes where

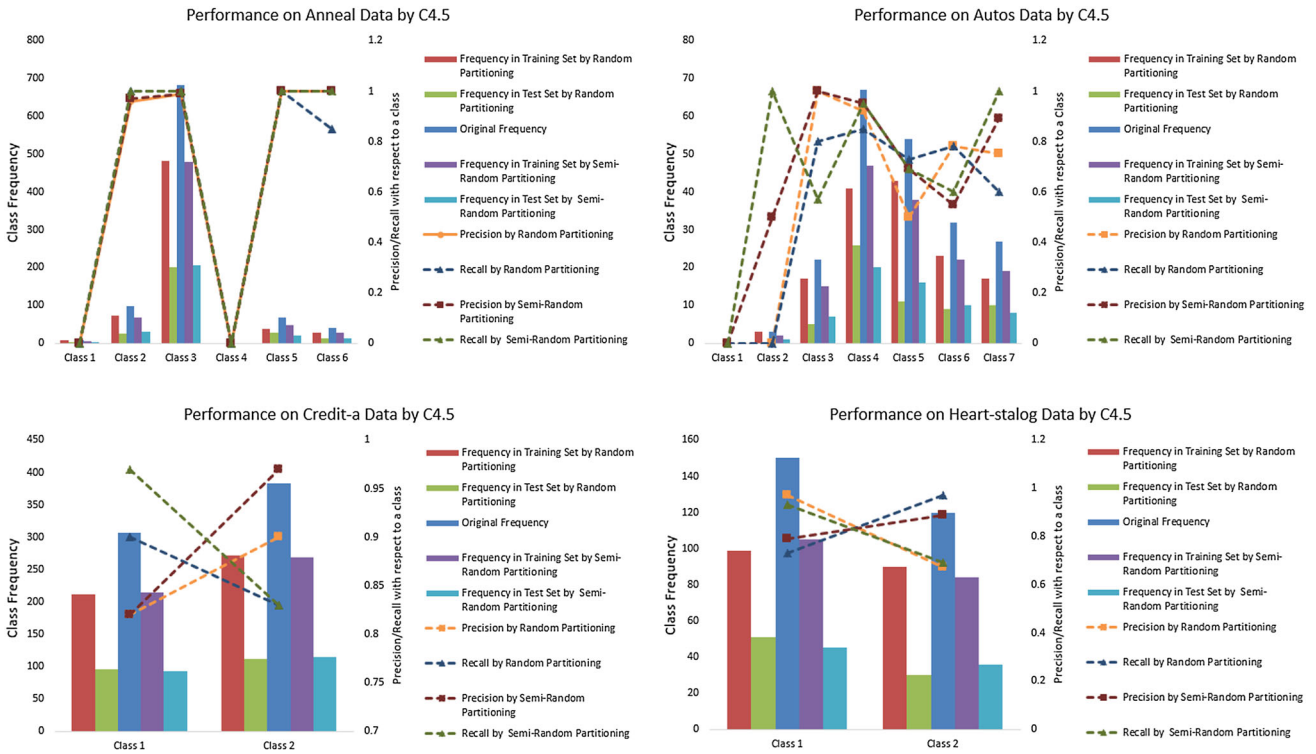

Fig. 3 Class distribution and performance (precision and recall) by C4.5 for random and semi-random partitioning for the 'anneal', 'autos', 'credit-a', and 'heart-statlog' data sets

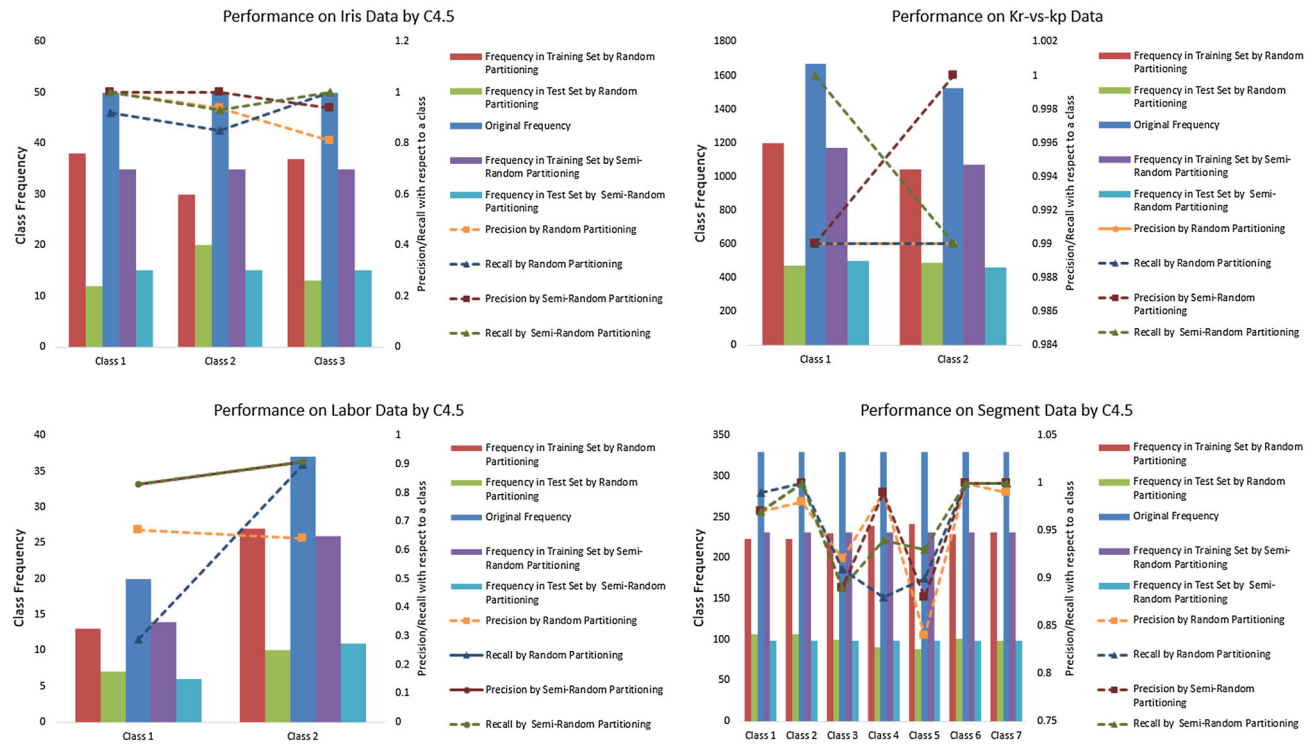

Fig. 4 Class distribution and performance (precision and recall) by C4.5 for random and semi-random partitioning for the 'iris', 'kr-vs-kp', 'labor', and 'segment' data sets

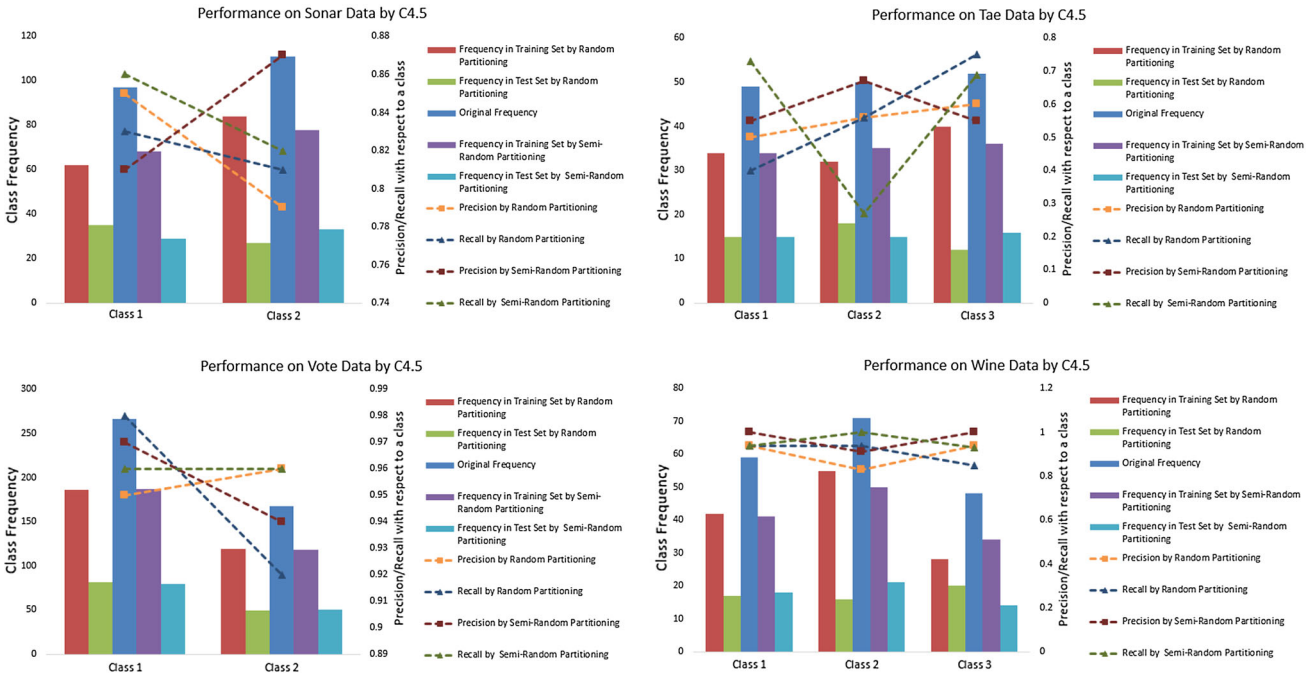

Fig. 5 Class distribution and performance (precision and recall) by C4.5 for random and semi-random partitioning for the 'sonar', 'tae', 'vote', and 'wine' data sets

the change results in more instances in the training set, the recall values are higher, while for the classes where the change results in more instances in the test set, the precision is higher; for the classes where there is little or no change, the difference in results may be due to sample representativeness. For the 'wine' data set, the random partitioning results in a more balanced distribution across classes in the training set, which may explain the better performance.

The accuracy for random and semi-random partitioning is the same on the 'sonar' data set, for which the random partitioning leads to a more imbalanced training set, with the same effect as above, i.e., when there are more instances in the training set, the recall is higher, while when there are more instances in the test set, the precision is higher.

For 6 data sets, i.e., 'anneal', 'autos', 'credit-a', 'iris', 'kr-vs-kp', and 'vote', the semi-random partitioning has up to 4% better accuracy than random partitioning. For the 'anneal', 'credit-a', and 'kr-vs-kp', the class distribution is very similar for random and semi-random partitioning—thus, the small difference is likely to be due to sample representativeness. For the 'autos', 'iris', and 'vote', the random partitioning leads to more class imbalance, which may affect the results.

The accuracy for the semi-random partitioning is higher than for the random one on three data sets, i.e., 'heart-statlog'(5%), 'labor' (12%), and 'tae' (18%). For the 'heart-statlog' and 'tae', the random partitioning leads to higher class imbalance in the training set, which may explain the results. For the 'labor' data set, the random partitioning leads to a better balance within the training set, but lower results than the semi-random partitioning which matches the original distribution—we believe that sample representativeness plays a big role in this situation and will investigate this in future work.

Table 13 and Figs. 9, 10, and 11 display the results for the experiments with K-nearest neighbour (K-NN) algorithm.

Similar to the results for Naive Bayes, we have four situations: (a) the accuracy for semi-random partitioning is slightly lower than for random partitioning, i.e., 'wine' (2%); (b) the two ways of partitioning have the same accuracy for the 'anneal' and 'labor' data set; (c) the semi-random partitioning has slightly better (up to 3%) accuracy, i.e., 'credit-a', 'iris', 'kr-vs-kp', 'segment', 'sonar', and 'vote'; (d) the accuracy is considerably higher (5% or more) for the semi-random partitioning, i.e., 'autos' (6%), 'heart-statlog', and 'tae'.

For the 'wine' data set, on which the random partitioning leads to 2% better accuracy, the partitioning leads to a higher number of instances in the training set for classes 2 and 3, which have the same or higher recall compared with semi-random partitioning. For class 1, there are more instances in the test set for the random partitioning, which has a higher precision than semi-random partitioning.

Table 12 NB performance on accuracy, precision, and recall

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Anneal								
R								
Precision	0.66	0.76	0.99	0.00	1.00	0.24		0.84
Recall	0.50	1.00	0.79	0.00	1.00	1.00		
SR								
Precision	0.50	0.79	0.99	0.00	1.00	0.30		0.86
Recall	1.00	1.00	0.82	0.00	1.00	0.92		
Autos								
R								
Precision	0.00	1.00	0.30	0.50	0.59	0.50	0.67	0.52
Recall	0.00	1.00	0.43	0.32	0.87	0.44	0.57	
SR								
Precision	0.00	1.00	0.42	0.80	0.55	0.20	0.67	0.53
Recall	0.00	1.00	0.71	0.40	0.69	0.20	0.75	
Credit-a								
R								
Precision	0.89	0.79						0.82
Recall	0.68	0.93						
SR								
Precision	0.91	0.78						0.83
Recall	0.67	0.95						
Heart-statlog								
R								
Precision	0.77	0.94						0.84
Recall	0.95	0.74						
SR								
Precision	0.86	0.94						0.89
Recall	0.96	0.81						
Iris								
R								
Precision	1.00	1.00	0.87					0.96
Recall	1.00	0.89	1.00					
SR								
Precision	1.00	1.00	1.00					1.00
Recall	1.00	1.00	1.00					
kr-vs-kp								
R								
Precision	0.87	0.89						0.88
Recall	0.91	0.85						
SR								
Precision	0.88	0.89						0.89
Recall	0.91	0.87						
Labor								
R								
Precision	0.75	0.92						0.88
Recall	0.75	0.92						
SR								
Precision	1.00	1.00						1.00
Recall	1.00	1.00						

Table 12 continued

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Segment								
R								
Precision	0.76	1.00	0.69	0.91	0.41	0.98	1.00	0.81
Recall	0.99	1.00	0.18	0.83	0.71	0.97	0.99	
SR								
Precision	0.79	1.00	0.57	0.90	0.43	0.95	1.00	0.80
Recall	0.97	1.00	0.12	0.87	0.68	0.97	1.00	
Sonar								
R								
Precision	0.77	0.79						0.77
Recall	0.89	0.60						
SR								
Precision	0.73	0.83						0.77
Recall	0.83	0.73						
Tae								
R								
Precision	0.63	0.50	0.25					0.47
Recall	0.80	0.26	0.36					
SR								
Precision	0.65	0.63	0.69					0.65
Recall	0.73	0.67	0.56					
Vote								
R								
Precision	0.96	0.81						0.90
Recall	0.88	0.93						
SR								
Precision	0.97	0.83						0.91
Recall	0.88	0.96						
Wine								
R								
Precision	1.00	1.00	1.00					1.00
Recall	1.00	1.00	1.00					
SR								
Precision	0.94	0.95	1.00					0.98
Recall	0.97	1.00	1.00					

For the 'anneal' data set, the class distribution is similar for random and semi-random partitioning, thus justifying the similar performance. For the 'labor' data set, the random partitioning leads to a better balance in the training set and a similar performance to the semi-random partitioning. This better class balance occurred also for the NB algorithm; however, the results were worst—the different results for the K-NN algorithms support our hypothesis that sample representativeness plays an important role in explaining these results.

When the semi-random partitioning leads to slight improvements in accuracy, i.e., 'credit-a', 'iris', 'kr-vs-kp', 'segment', 'sonar', and 'vote', we notice similar patterns: (1) for similar distributions, i.e., 'kr-vs-kp' and 'vote', the difference is likely to be due to sample representativeness; (2) when the random sampling leads to changes in the class distribution, an increase in the number of instances in the training set is associated with increase in recall, while the increase in the number of instances in the test set is associated with an increase in precision.

For the data sets with considerably higher accuracy for semi-random partitioning, there are two situations: (a) the class distribution is the same, i.e., 'heart-statlog'—consequently, the difference in results is probably due to sample representativeness; (b) the random partitioning leads to higher imbalance for some classes, which together with the

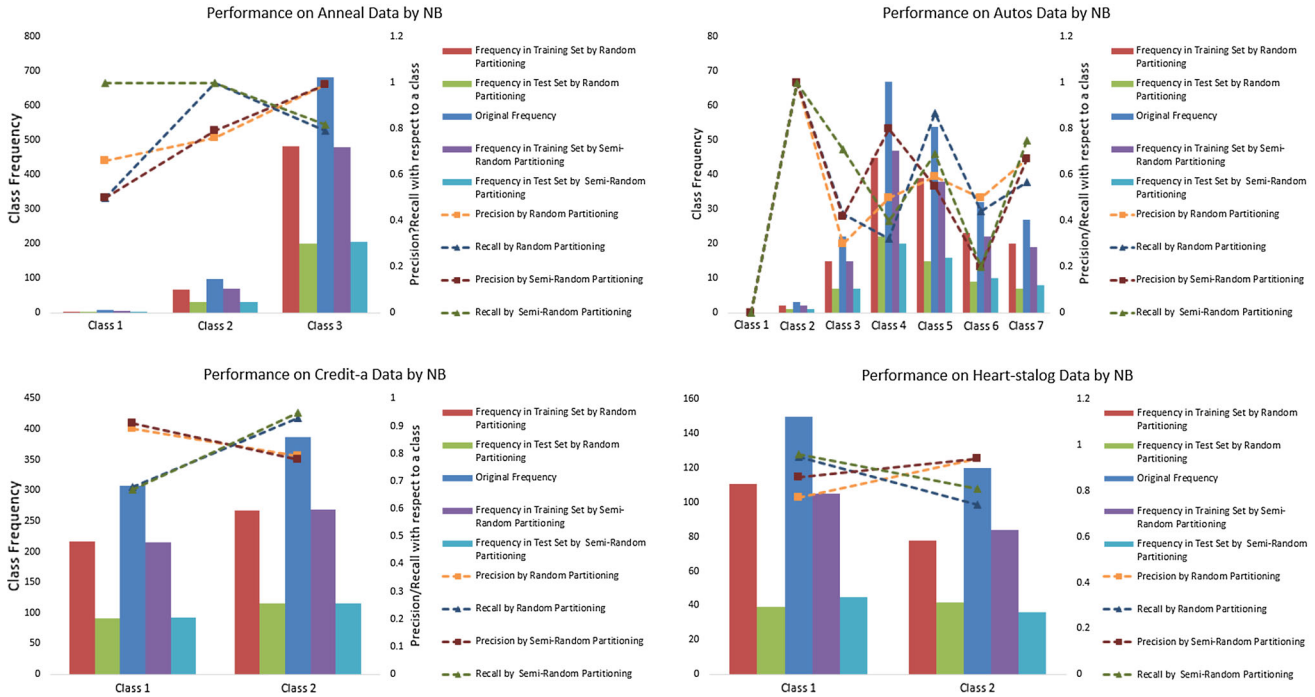

Fig. 6 Class distribution and performance (precision and recall) by NB for random and semi-random partitioning for the 'anneal', 'autos', 'credit-a', and 'heart-statlog' data sets

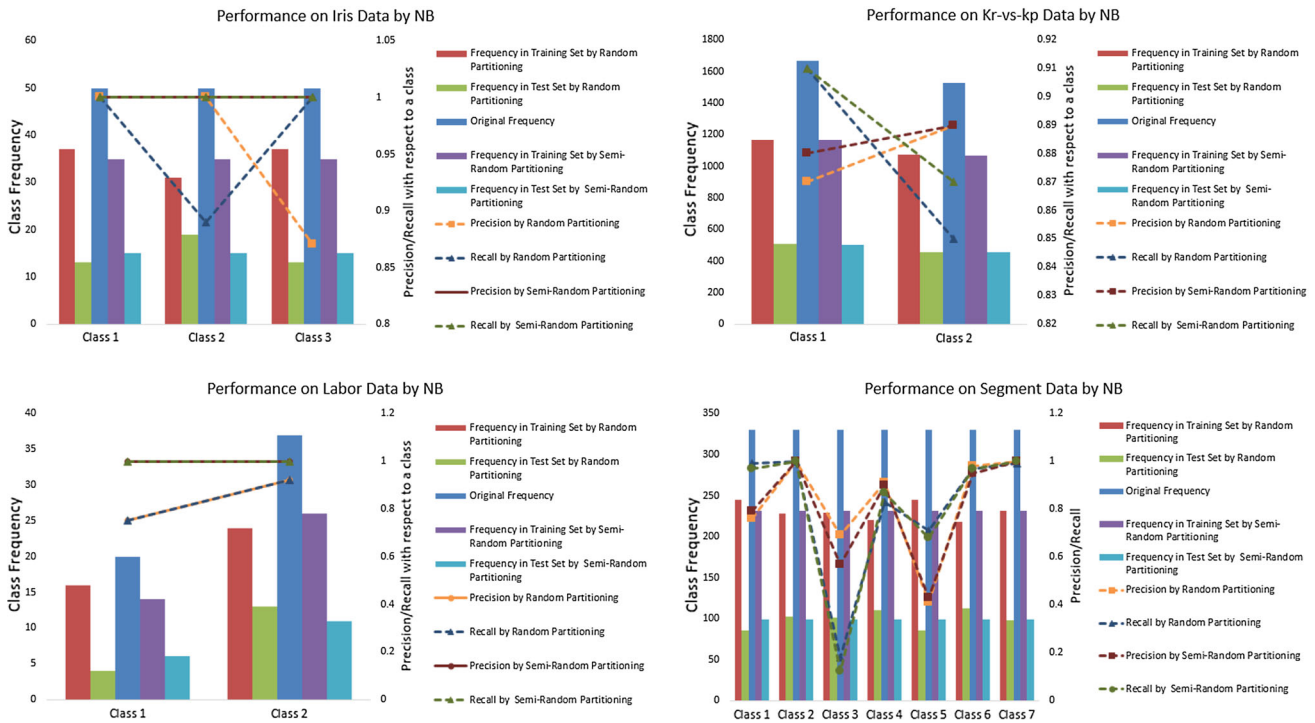

Fig. 7 Class distribution and performance (precision and recall) by NB for random and semi-random partitioning for the 'iris', 'kr-vs-kp', 'labor', and 'segment' data sets

sample representativeness explain the results, i.e., 'autos' and 'tae'.

To summarise, we noticed that the distribution of classes within the training and test sets has an effect on the performance results. In particular, there is an association between a larger number of instances in the training set and a higher recall and between a larger number of instances in the test set and a higher precision. A higher number of

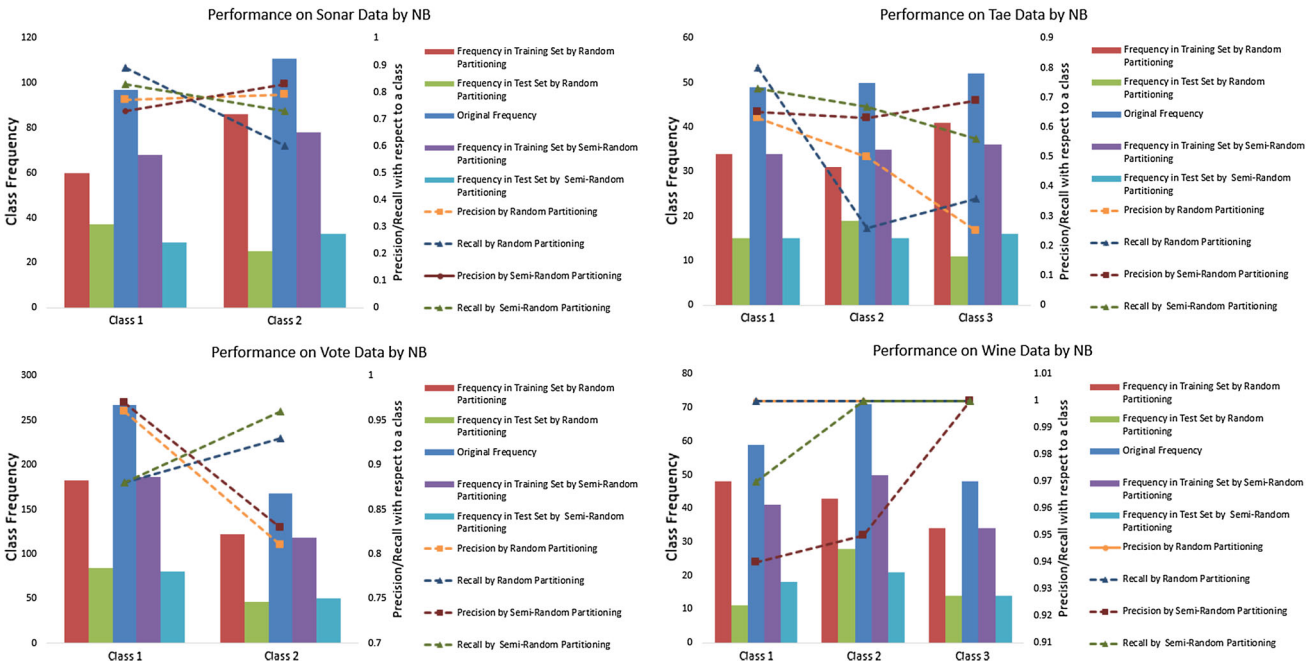

Fig. 8 Class distribution and performance (precision and recall) by NB for random and semi-random partitioning for the 'sonar', 'tae', 'vote', and 'wine' data sets

instances in the training set can mean more opportunities for learning, and, thus, a better knowledge of a particular class, which explains the higher recall. For a good performance, however, recall needs to be balanced with precision, i.e., ensure that the model can distinguish a particular class from the other classes; in other words, a low precision means that instances of a particular class is wrongly labeled with another class(es). This is more likely to be influenced by the distribution among classes, than the distribution of a class between the training and the test set, as the balance between classes in the training set has an influence on the capacity to learn to distinguish between classes (which is why class imbalance is known to lead to poor performance). This is supported by the fact that the semi-random partitioning results are more balanced in terms of precision and recall, while the random partitioning with imbalanced class distribution in the training set, as well as imbalance across the training and test sets, tend to have one of two combinations: (a) high precision and low recall, or (b) low recall and high precision.

The results also indicate that the class distribution within the training set has more influence on the performance than the class distribution within the test set. On the other hand, the distribution within the test set still requires consideration to accurately assess the performance of a model. For example, a small test sample may not sufficiently test the knowledge learned for a particular class—in an extreme situation, it may mean that knowledge is not tested at all. These aspects can be easily controlled with our proposed partitioning method.

Overall, the experimental results indicate that the adoption of the strategy of semi-random data partitioning involved in Level 2 of the multi-granularity framework proposed in Sect. 3 achieves effective control of the selection of training/test instances, towards avoiding the case of class imbalance in both training and test sets, especially when data sets are originally balanced or slightly imbalanced.

Our results also showed situations when the random and semi-random partitioning led to the same distribution, but different results. We believe that these are likely to be explained by the sample representativeness issues, which we will address in future work with experiments on Level 3 of the propose multi-granularity framework.

5 Conclusions

In this paper, we identified two issues resulting from the operation of random partitioning of data into a training set and a test. In particular, we argued that a fully random way of data partitioning could lead to the case of class imbalance and to sample representativeness issues, i.e., the case that training instances are highly dissimilar to the test instances. To address these issues, we proposed a multi-granularity framework for semi-random data partitioning. The proposed framework involves both granulation and organization in the setting of granular computing, towards more effective data partitioning in a semi-random way.

Table 13 K-NN performance on accuracy, precision, and recall

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Anneal								
R								
Precision	0.67	0.95	0.97	0.00	1.00	0.89		0.96
Recall	0.50	1.00	0.98	0.00	1.00	0.62		
SR								
Precision	1.00	0.90	0.99	0.00	1.00	0.69		0.96
Recall	1.00	0.93	0.96	0.00	1.00	0.92		
Autos								
R								
Precision	0.00	0.00	0.67	0.60	0.65	0.29	1.00	0.52
Recall	0.00	0.00	0.33	0.67	0.69	0.36	0.30	
SR								
Precision	0.00	0.00	0.71	0.58	0.55	0.50	0.67	0.58
Recall	0.00	0.00	0.71	0.55	0.75	0.40	0.50	
Credit-a								
R								
Precision	0.85	0.88						0.87
Recall	0.82	0.90						
SR								
Precision	0.91	0.88						0.89
Recall	0.84	0.93						
Heart-statlog								
R								
Precision	0.71	0.70						0.70
Recall	0.77	0.62						
SR								
Precision	0.84	0.88						0.85
Recall	0.91	0.79						
Iris								
R								
Precision	1.00	0.93	0.87					0.93
Recall	1.00	0.88	0.92					
SR								
Precision	1.00	0.88	1.00					0.96
Recall	1.00	1.00	0.87					
kr-vs-kp								
R								
Precision	0.93	0.98						0.95
Recall	0.98	0.92						
SR								
Precision	0.94	0.97						0.96
Recall	0.97	0.94						
Labor								
R								
Precision	1.00	0.92						0.94
Recall	0.80	1.00						
SR								
Precision	1.00	0.92						0.94
Recall	0.83	1.00						

Table 13 continued

Data set	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Accuracy
Segment								
R								
Precision	0.98	1.00	0.87	0.94	0.80	0.98	1.00	0.94
Recall	0.97	1.00	0.91	0.88	0.82	1.00	0.99	
SR								
Precision	0.96	1.00	0.85	0.99	0.87	0.96	1.00	0.95
Recall	0.98	1.00	0.95	0.86	0.83	1.00	1.00	
Sonar								
R								
Precision	0.86	0.74						0.79
Recall	0.73	0.86						
SR								
Precision	0.84	0.78						0.81
Recall	0.72	0.88						
Tae								
R								
Precision	0.44	0.36	0.50					0.44
Recall	0.50	0.27	0.57					
SR								
Precision	0.54	0.57	0.63					0.59
Recall	0.47	0.53	0.75					
Vote								
R								
Precision	0.97	0.87						0.93
Recall	0.91	0.96						
SR								
Precision	0.96	0.90						0.94
Recall	0.94	0.94						
Wine								
R								
Precision	1.00	1.00	0.92					0.98
Recall	1.00	0.93	1.00					
SR								
Precision	0.95	1.00	0.93					0.96
Recall	1.00	0.90	1.00					

We conducted several experiments using 12 UCI data sets and three popular machine learning algorithms (C4.5, Naive Bayes, and K-nearest neighbour). We focused on Level 2 of the framework for avoiding class imbalance. The results show interesting effects of the class distribution within the training and test sets on overall accuracy, as well as precision and recall per class. The results have also shown that the same class distribution for random and semi-random partitioning can lead to different performance results—we believe that this is most likely due to the issues of sample representativeness, which are addressed in Level 3 of the proposed framework.

In particular, for Level 3, we argued the necessity that each class of instances needs to be specialized into a number of subclasses, by grouping instances from the same class based on their similarity. By sampling data for the training and test sets at the level of these subclasses, the sample representativeness can be controlled across both the training and test sets, thus avoiding situations in which knowledge is learned but not tested, or knowledge that is tested without having been learned.

In this paper, we focused on the preservation of the original class distribution within the training and test sets. While this approach is suitable for balanced and slightly

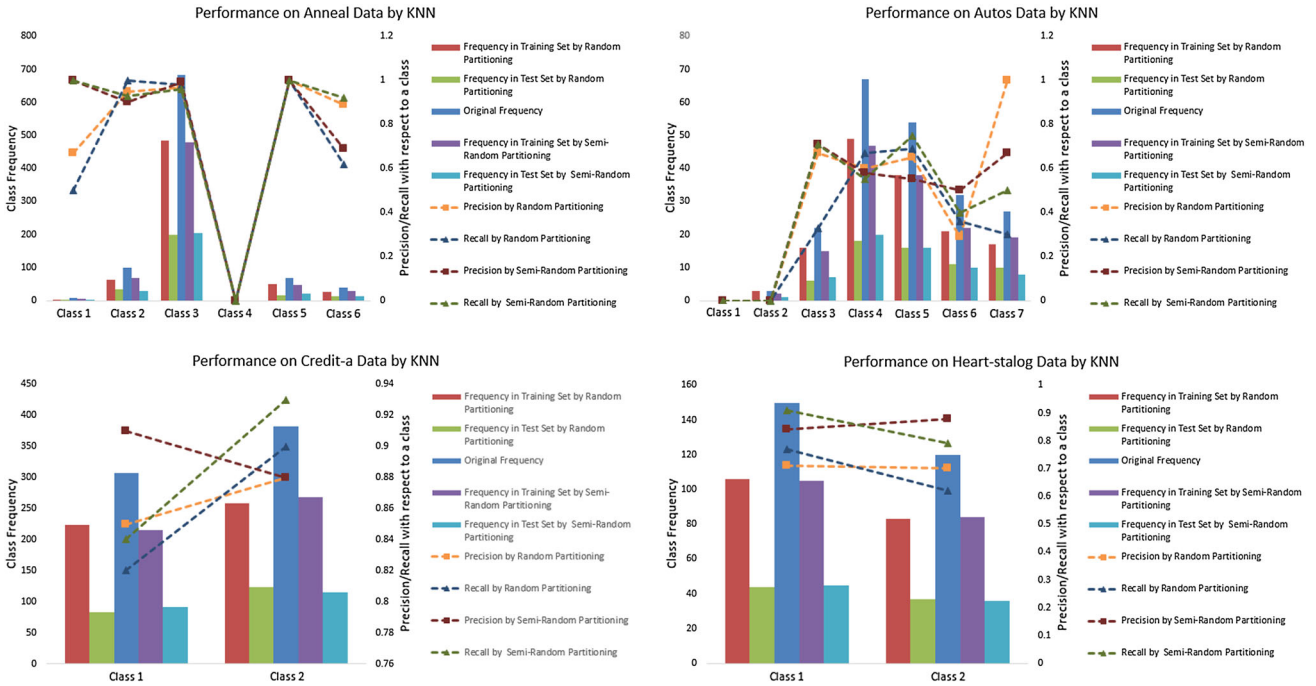

Fig. 9 Class distribution and performance (precision and recall) by K-NN for random and semi-random partitioning for the 'anneal', 'autos', 'credit-a', and 'heart-statlog' data sets

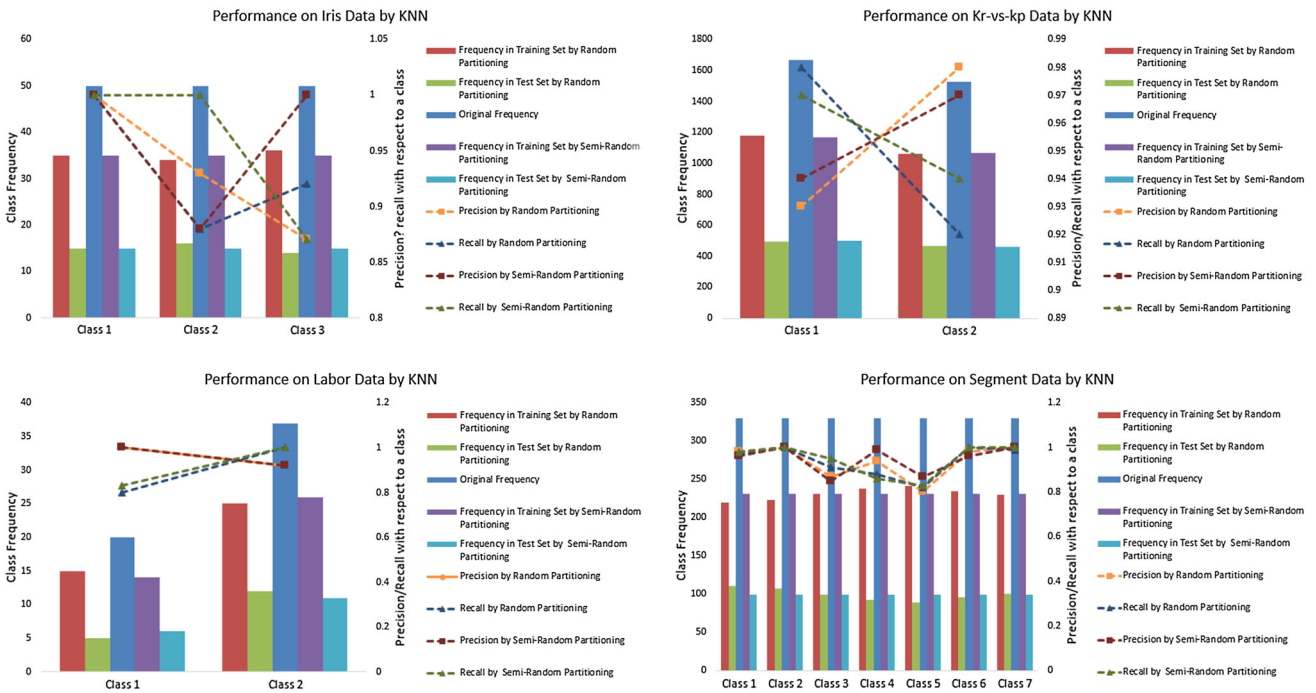

Fig. 10 Class distribution and performance (precision and recall) by K-NN for random and semi-random partitioning for the 'iris', 'kr-vs-kp', 'labor', and 'segment' data sets

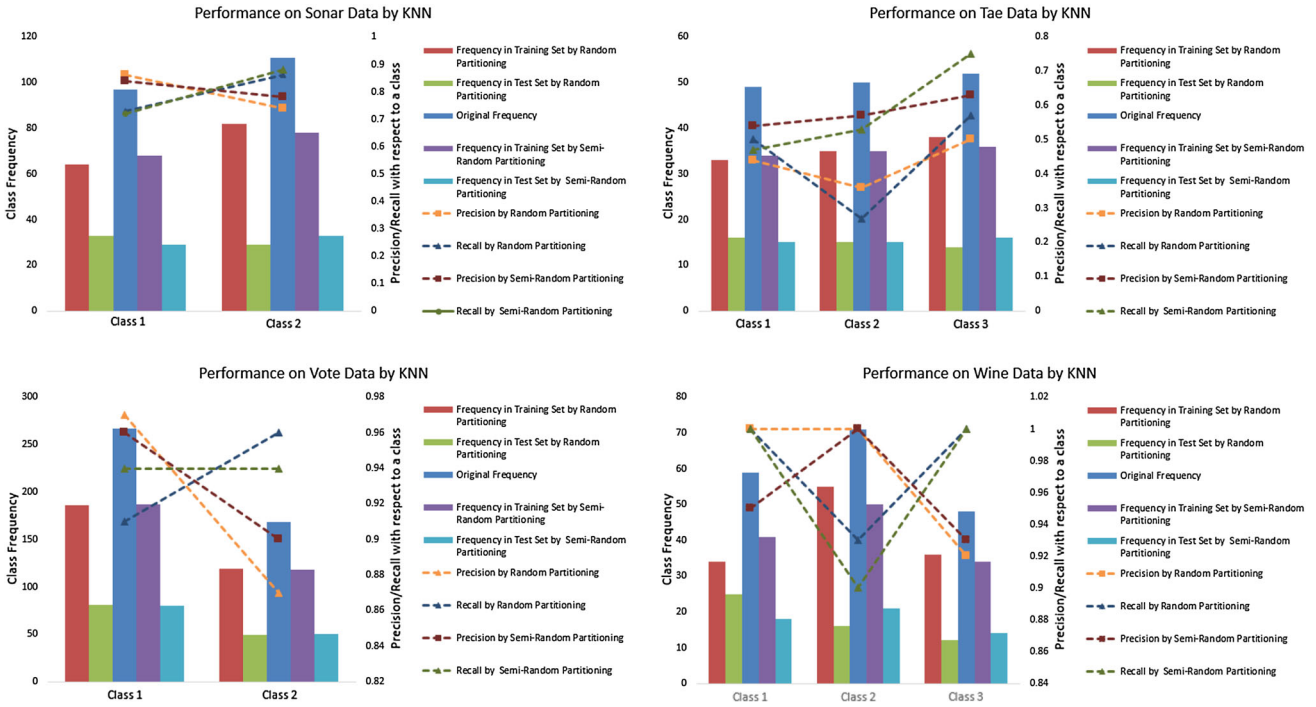

Fig. 11 Class distribution and performance (precision and recall) by K-NN for random and semi-random partitioning for the 'sonar', 'tae', 'vote', and 'wine' data sets

imbalanced data sets, it may not be the best for highly imbalanced data sets. In future work, we will investigate how the principles of Level 2 in our framework can be adapted for imbalanced data sets, using stratified sampling (mentioned in Sect. 3.2) to achieve a better balance for the class distribution, particularly in the training set.

Acknowledgements The authors acknowledge support for the research reported in this paper through the Research Development Fund at the University of Portsmouth.

References

Ali A, Shamsuddin SM, Ralescu AL (2015) Classification with class imbalance problem: a review. Int J Adv Soft Comput Appl 7(3):176–204

Antonelli M, Ducange P, Lazzerini B, Marcelloni F (2016) Multi-objective evolutionary design of granular rule-based classifiers. Granul Comput 1(1):37–58

Devijver PA (1982) Pattern recognition: a statistical approach. Prentice-Hall, London

Dubois D, Prade H (2016) Bridging gaps between several forms of granular computing. Granul Comput 1(2):115–126

Esfahani MS, Dougherty ER (2014) Effect of separate sampling on classification accuracy. Bioinformatics 30(2):242–250

Geisser S (1993) Predictive inference. Chapman and Hall, New York

Hu H, Shi Z (2009) Machine learning as granular computing. IEEE International Conference on Granular Computing. Nanchang, Beijing, pp 229–234

Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann, San Mateo, pp 1137–1143

Kreinovich V (2016) Solving equations (and systems of equations) under uncertainty: how different practical problems lead to different mathematical and computational formulations. Granul Comput 1(3):171–179

Lang K, Liberty E, Shmakov K (2016) Stratified sampling meets machine learning. In: Proceedings of the 33rd International Conference on Machine Learning. JMLR.org, New York, pp 2320–2329

Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml

Liu H, Cocea M (2017a) Granular computing based approach for classification towards reduction of bias in ensemble learning. Granul Comput 2(3)

Liu H, Cocea M (2017b) Fuzzy information granulation towards interpretable sentiment analysis. Granul Comput 3(1) (**in press**)

Liu H, Gegov A, Cocea M (2016a) Nature and biology inspired approach of classification towards reduction of bias in machine learning. International Conference on Machine Learning and Cybernetics. Jeju Island, South Korea, pp 588–593

Liu H, Gegov A, Cocea M (2016b) Rule based systems: a granular computing perspective. Granul Comput 1(4):259–274

Liu H, Gegov A, Cocea M (2016c) Rule based systems for big data: a machine learning approach. Springer, Switzerland

Liu H, Gegov A, Cocea M (2017) Unified framework for control of machine learning tasks towards effective and efficient processing

of big data. In: Data Science and Big Data: An Environment of Computational Intelligence. Springer, Switzerland, pp 123–140
Livi L, Sadeghian A (2016) Granular computing, computational intelligence, and the analysis of non-geometric input spaces. Granul Comput 1(1):13–20
Longadge R, Dongre SS, Malik L (2013) Class imbalance problem in data mining: review. Int J Comput Sci Netw 2(1):83–87
Merriam-Webster (2016) http://www.merriam-webster.com/
Min F, Xu J (2016) Semi-greedy heuristics for feature selection with test cost constraints. Granul Comput 1(3):199–211
Pedrycz W (2011) Information granules and their use in schemes of knowledge management. Sci Iran 18(3):602–610
Pedrycz W, Chen S-M (2011) Granular computing and intelligent systems: design with information granules of higher order and higher type. Springer, Heidelberg
Pedrycz W, Chen S-M (2015a) Granular computing and decision-making: interactive and iterative approaches. Springer, Heidelberg
Pedrycz W, Chen S-M (2015b) Information granularity, big data, and computational intelligence. Springer, Heidelberg
Peters G, Weber R (2016) Dcc: a framework for dynamic granular clustering. Granul Comput 1(1):1–11
Quinlan RJ (1993) C4.5: programs for machine learning. Morgan Kaufmann Publishers, San Francisco
Rish I (2001) An empirical study of the Naive Bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence, vol 3, no. 22, pp 41–46
Skowron A, Jankowski A, Dutta S (2016) Interactive granular computing. Granul Comput 1(2):95–113
Sotiropoulos DN, Tsihrintzis GA (2017) The class imbalance problem. Springer, Cham
Srndal C-E, Swensson B, Wretman J (1992) Model assisted survey sampling. Springer, New York
Wilke G, Portmann E (2016) Granular computing as a basis of humandata interaction: a cognitive cities use case. Granul Comput 1(3):181–197
Yao Y (2005) Perspectives of granular computing. In: Proceedings of 2005 IEEE International Conference on Granular Computing. Beijing, China, pp 85–90
Zadeh L (2015) Fuzzy logic: a personal perspective. Fuzzy Sets Syst 281:4–20